

De Nouvelles Architectures pour les Communications 1996

**Méthode d'interaction asynchrone
dans un environnement X-Window**

L. Gannoun, P. Dubois, J. Labetoulle

Eurécom, Sophia Antipolis

être synchronisés et présents durant toute la session de travail coopératif. Les systèmes de partage d'application peuvent être plus efficaces si le mode de travail peut être étendu à une interaction asynchrone qui ne contraint pas les participants d'être en-ligne en même temps.

Ce besoin est à l'origine de ce travail à l'Institut Eurécom. Dans cet Institut une plate-forme de télé-enseignement BETEL (Broad Band Exchange over Trans-European Links) a été développée. Une application de télé-enseignement est construite sur cette plate-forme et testée entre Eurécom et l'Ecole Polytechnique Fédérale de Lausanne (EPFL). Cette application utilise des outils de vidéoconférence et de partage d'espace de travail pour permettre une interaction entre un professeur à l'EPFL et un groupe d'étudiants à Eurécom. Le professeur situé dans un site peut enseigner une classe située dans un autre site et superviser les étudiants dans leurs travaux individuels (Pusztaszeri, Biersack, Dubois, Goud, Gros & Hubaux, 1994; Gros, 1993; Marom & Gros 1993). Cependant, BETEL est limité car il ne supporte pas l'enseignement et l'apprentissage en temps différé. Si le professeur n'est pas en-ligne en même temps avec les étudiants il ne peut pas suivre et corriger leurs travaux.

L'objectif de notre travail est de concevoir et de développer une méthode qui offre la possibilité d'une interaction asynchrone dans un environnement X-window. Cette méthode va être utilisée durant une session de télé-enseignement entre un professeur et des étudiants. Dans notre travail actuel, nous proposons de partager l'interface utilisateur graphique (GUI) d'une application X-window. Par conséquent ce système doit dans une première étape offrir la possibilité aux étudiants d'enregistrer leurs sessions-X et de les commenter. Dans une deuxième étape permettre à un professeur distant de visualiser et de rejouer la session enregistrée puis de la corriger. Ces capacités doivent être implémentées d'une façon distribuée pour permettre une collaboration interactive à distance.

Le papier est organisé comme suit: nous commençons par présenter les modèles de base des systèmes de partage d'application. Ensuite, nous présentons les services multimédia inhérents à la régénération d'une session d'une application-X. Après nous articulons avec la présentation de notre architecture d'interaction asynchrone utilisant les services de régénération d'une session. Enfin, nous terminons par une conclusion et des remarques et nous exposons nos prochains travaux.

2. Les Systèmes de partage d'application

Un système de partage d'application fournit un ensemble de mécanismes pour permettre à des membres d'une communauté coopérative, chacun sur sa station de travail, de partager simultanément des informations et d'interagir avec une même application-X. Afin d'appréhender le fonctionnement de ces systèmes, nous exposons les mécanismes d'exécution d'une application-X.

2.1 Le système X-window

Le système X-window est un système de gestion de l'interface utilisateur (User Interface Management System, UIMS) qui fournit les capacités de créations simples d'interfaces utilisateurs

2.2 Modèle à partage d'événements

Le modèle de partage d'événements considère que le même client-X est localement exécuté par tous les participants à la session. Dans ce modèle, le Système de Partage d'Application-X (SPAX) transporte les événements d'un utilisateur qui a la permission (possède un jeton) à toutes les applications et bloque les événements provenant des autres utilisateurs. Par conséquent chaque session d'un participant devra avoir sa copie de l'application-X à partager avec ses données mise à jour. La figure 2 représente l'architecture d'un SPAX basé sur le partage d'événements.

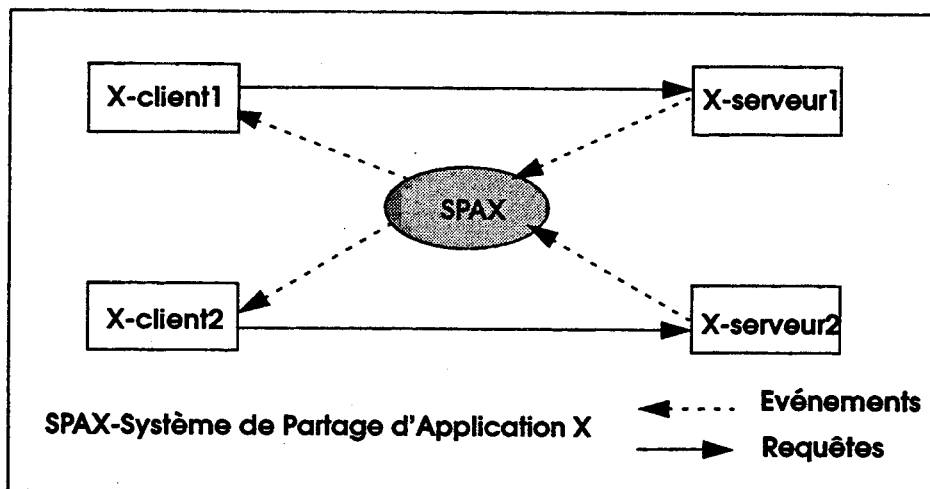


Figure 2. Modèle à partage d'événements

Minenko (95) énumère les inconvénients majeurs de ce modèle:

- Le partage de l'application est basé sur l'hypothèse du comportement déterministe d'une application.
- L'environnement d'application doit être identique pour tous les participants.
- L'application à partager doit être disponible pour chaque participant.
- L'impossibilité de maintenir une copie consistante entre les copies multiples de l'application. Par exemple, la perte des événements pour une copie d'application (si les événements ne sont pas acceptés) jette cette copie en dehors du processus de synchronisation avec les autres. Ces inconvénients sérieux rendent le modèle à partage d'événements inexploitable ce qui a poussé à la conception d'un autre modèle plus consistant.

2.3 Modèle à partage de requêtes

Le modèle à partage de requêtes utilise une seule copie d'application-X qui est démarrée sur un site. Les sorties graphiques de cette application (requêtes) sont multiplexées par le SPAX à tous les serveurs-X des autres sites. Les événements provenant de ces serveurs-X sont collectés et

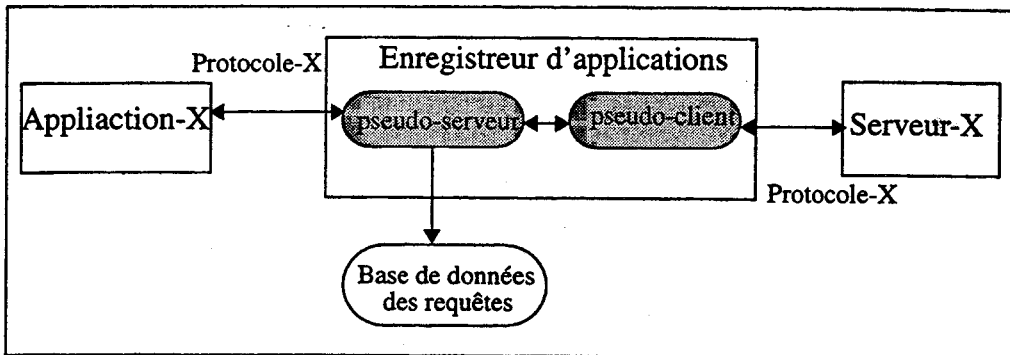


Figure 4. Architecture d'enregistrement de l'application

L'enregistreur d'application comporte deux composantes:

- Le pseudo serveur: Ce composant intercepte la connexion de l'application-X au serveur puis, capture les requêtes et les stocke dans une base de données. La politique de stockage des requêtes dépend du modèle de l'application et sera discuté dans le prochain paragraphe. Le pseudo-serveur initie la connexion à l'application-X et passe les requêtes au pseudo-client. Il reçoit encore du pseudo-client les événements, les réponses et les erreurs et les passe à l'application. Le pseudo serveur filtre aussi les requêtes et ne stocke pas celles qui nécessitent des réponses. Ces requêtes dernières n'affectent pas l'interface utilisateur graphique et par conséquent ne participent pas à la régénération de la session de l'application.

- Le pseudo client: joue le rôle d'un client ordinaire, il initie une connexion au serveur et lui passe les requêtes provenant du pseudo-serveur. Une fois qu'il reçu les événements, les réponses et les erreurs du serveur il les passent au pseudo-serveur.

L'enregistreur d'application filtre et stocke les requêtes dans une base de données. Ces requêtes sont stockées sans aucune modification et conformément à une politique de stockage.

3.2 Politique de stockage

La politique de stockage des requêtes dépend du modèle de l'application-X qui comporte deux modèles globaux.

- Le modèle à processus unique et,
- Le modèle à processus multiples

Une application-X avec un modèle à processus unique est associée à un unique processus client-X. L'organisation des requêtes est alors simple. Les requêtes sont stockées séquentiellement, et doivent être estampillées (timestamped), pour pouvoir rejouer l'application avec la même perception que l'utilisateur a eu de son interface durant l'exécution de l'application. En l'occur-

requêtes qui appartiennent à un seul client-X sont stockées dans la même file de requêtes. Les précédences entre les requêtes sont résolues par les estampilles. Un simple algorithme qui fait une régénération efficace d'une application-X est celui d'une simulation distribuée optimiste. Chaque file de requêtes associée à un client-X est considérée comme un canal d'entrée. Ces différents files de requêtes sont considérés comme des canaux d'entrées à l'ordonnanceur.

L'ordonnanceur dispose d'une variable globale (GroupReq) qui représente le numéro du groupe des requêtes à lancer au serveur. Cet ordonnanceur charge le groupe de requêtes parmi les groupes qui sont en tête des canaux d'entrée et dont le numéro, correspond à la valeur de la variable GroupReq. Il calcule Alors le temps d'attente (TimeOut) associée à ce groupe de requêtes pour le lancer au serveur. Une fois le temps d'attente est écoulé, le groupe de requêtes est lancé. Ensuite, l'ordonnanceur incrémente la variable GroupReq pour charger le groupe de requêtes suivant.

Chaque file de requêtes est associée à une connexion particulière au serveur-X. Un groupe de requêtes sélectionné est lancé à travers une connexion qui correspond à sa file de requêtes. Les estampilles de requêtes servent à calculer le délai d'attente avant le lancement des requêtes. Ce délai peut être modifié pour permettre une régénération flexible de l'application-X.

3.5 Architecture d'un pseudo-client pour la régénération

Un pseudo-client pour la régénération est un client virtuel qui fournit le Service de Régénération de l'Application (SRA). Ce service permet à l'utilisateur de régénérer une session enregistrée d'application-X. Le client virtuel est substitué à l'application-X réelle et lance convenablement les requêtes stockées. Ses tâches principales sont: le chargement et l'ordonnement des requêtes, translation des requêtes et lancement des requêtes et gestion des connexions au serveur. La figure 5 montre l'architecture d'un pseudo-client pour la régénération.

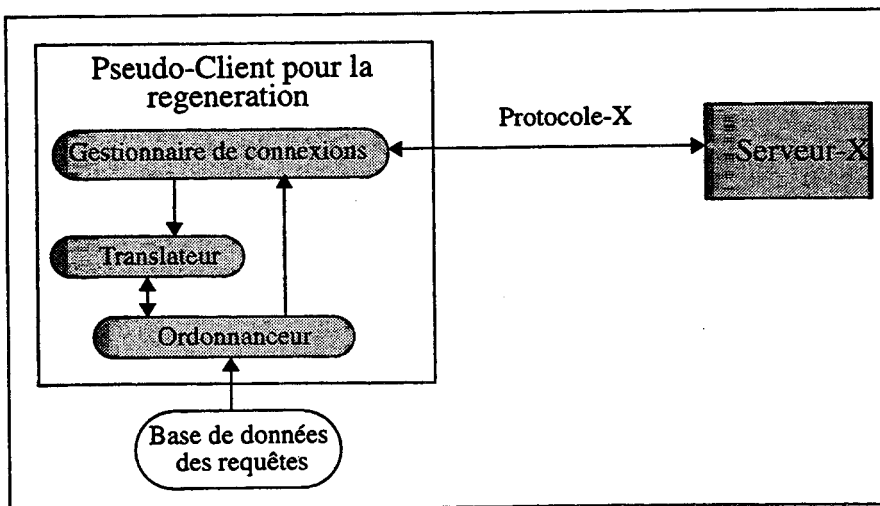


Figure 5. Architecture d'un Pseudo-client pour la régénération

4.2 Architecture globale d'interaction basée sur le WWW

Nous présentons dans ce qui suit notre architecture d'interaction qui utilise le Web (WWW). Nous dénotons par profile de session-X à la fois les informations stockées concernant la session-X et les commentaires textuels et les questions de l'étudiant. La figure 6 montre l'architecture d'interaction qui permet une communication entre l'étudiant et le professeur.

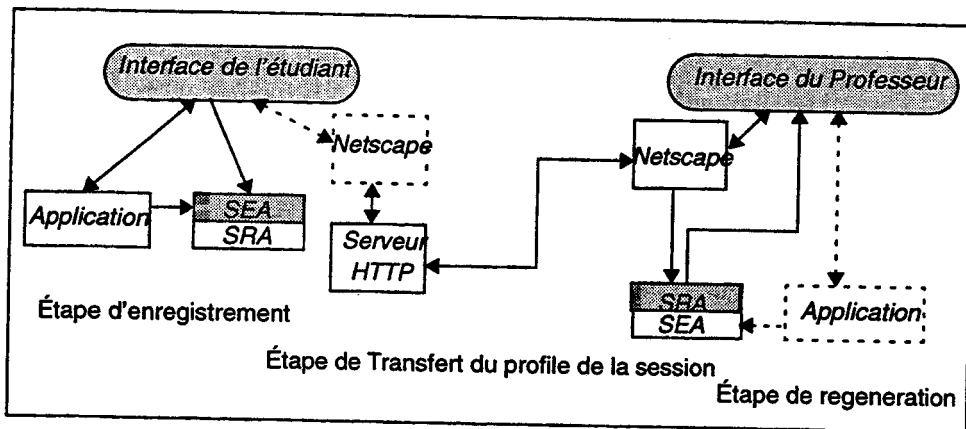


Figure 6. Architecture d'interaction basée sur le WWW

Trois étapes sont distinguées :

Étape d'enregistrement : Durant cette étape la session-X de l'étudiant est stockée. Les commentaires et questions de l'étudiant sont aussi stockés et représentés par des documents HTML. Les relations entre les portions de la session et ces documents HTML sont établis. Tous ces informations constituent un profile de la session de l'étudiant.

Étape de transfert du profile de la session : Cette étape est effectuée plus tard par le professeur qui répond à l'invitation de l'étudiant. Cette invitation est simplement effectuée par un message mail dans lequel l'étudiant donne au professeur l'adresse URL (Uniform Resource Locator) du profile de la session-X. En utilisant cette adresse le professeur peut charger le profile de la session et commencer par initier la régénération de la session de l'étudiant.

Étape de régénération : Dans cette étape le professeur régénère la session de l'étudiant. La régénération est interrompu par des points d'arrêts. Ces points permettent d'afficher les commentaires et les questions de l'étudiant. Après avoir lu ces commentaires le professeur continue à régénérer la session-X. Dans le but de corriger le travail de l'étudiant, le professeur peut lui notifier des indications. Ces indications sont soit textuelles soit des sessions-X de démonstrations qui montrent à l'étudiant comment il faut procéder pour résoudre les problèmes rencontrés. Ceci nécessite la commutation entre une étape de régénération et une autre d'enregistrement. Afin de construire un profile de session-X pour une réponse efficace, nous devons avoir les deux étapes d'enregistrement et de régénération en parallèle. Ceci permet au professeur de faire des indications sur la session-X de l'étudiant.

Pour avoir la réponse, l'étudiant charge le profile de la session-X du professeur. L'adresse qui référence ce profile est fournie à l'étudiant par un message mail du professeur. L'étudiant com-

- de l' *International Workshop on Advanced Communications and Applications for High Speed Networks: IWACA'92* (pp. 401 - 405) Munchen.
- Chung, G., Jeffay, K., & Abdel-Wahab, H. M. (1993). Accommodating late-comers in shared window system. *IEEE Computer*, 26(1) pp. 72-74.
- Dermler, G., Gutekunst, T., Plattner, B., Ostrowski, E., Ruge, F., & Weber, M. (1993). Constructing a distributed multimedia joint viewing and tele-operation service for heterogeneous workstation environments. *Proceedings, de la fourth IEEE Workshop on Future Trends of Distributed Computing Systems* (pp. 8 - 15). Lisbon.
- Frivold, T. J., Lang, R. E., & Fong, M. W. (1994). Extending WWW for synchronous collaboration. *Electronic Proceedings du Second World Wide Web Conference' 94: Mosaic and the Web* [On-line]. Available: <http://www.ncsa.uiuc.edu/SDG/IT94/Proceedings/CSCW/frivold/frivold.html>.
- Gannoun, L., Dubois, P., Labetoulle, J. Recording and Replaying of an X-session: An approach for an Asynchronous Interaction Method. *Proceedings of the World Conference on Educational Multimedia and Hypermedia & World Conference on Educational Telecommunications, ED-MEDIA/ED-TELECOM'96*, June 17- 22, 1996 Boston, Mass.-USA.
- Gannoun, L., Dubois, P., Labetoulle, J. Asynchronous Interaction Method for a Remote Teleteaching Session. *Accepté pour publication dans l'International Journal of Educational Telecommunication (IJET)*. AACE publications.
- Gros, P., (1993). Remote teaching application: Dialog states and transitions, connection. *Rapport Technique 1-EURE-008*. Institut Eurecom, France, 1993.
- Gutekunst, T., & Plattner, B. (1993). Sharing multimedia applications among heterogeneous workstation. *Proceedings, Second International Conference on Broad Band Islands 1993* (pp. 173 - 191). Athens.
- Gutekunst, T., Bauer, D., Caronni, G., Hasan, & Plattner, B. (1995, February). A distributed and policy-free general-purpose shared window system. *IEEE/ACM Transactions on Networking*.
- Lauwers, J. C., Lantz, K. A., (1990). Collaboration awareness in support of collaboration transparency: requirements of the next generation of shared window systems. *Proceedings de ACM CHI'90 Conference (Human Factors in Computing Systems)*, pp. 303 -311. Seattle.
- Marom, R., & Gros, P. (1993). Remote teaching application ergonomics/U.I specifications. *Rapport Technique 1-EURE-006*. Institut Eurecom, France.